Using Internal Validity Measures to Compare Clustering Algorithms

Toon Van Craenendonck
KU Leuven
Department of Computer Science
toon.vancraenendonck@cs.kuleuven.be

Hendrik Blockeel
KU Leuven
Department of Computer Science
hendrik.blockeel@cs.kuleuven.be

Abstract

An obvious question to ask yourself when you want to cluster a data set is: which algorithm should I use? Given the variety of existing clustering algorithms, answering this question is far from trivial. A straightforward strategy is to simply run several algorithms, with a number of different parameter configurations, and afterwards select the best clustering from the generated set of solutions. But how do we select the best one? One way to do this is by using internal validity measures, which map a clustering to a number indicating its quality. For this strategy to be valid, we need an internal measure that allows for a fair comparison between clustering algorithms. We have experimented with four of these validity measures and six clustering algorithms. We observed some undesired properties for each of the measures, making them unsuitable for such a comparison.

1. Introduction

Jain [7] defines clustering as the task of organizing data into sensible groups. Many other definitions can be found in the literature. Likewise, many different clustering algorithms exist, which may all produce very different partitions of the same data set. Even a single clustering algorithm can yield wildly different results depending on the chosen parameters. Consequently, a common strategy to cluster a particular data set is to try out several algorithms and parameter configurations. The user is then left with the task of selecting the "best" clustering from the resulting set of solutions. This is a difficult task, as there is no consensus on what a "good" clustering exactly looks like [1]. Evaluating clusterings is more challenging than evaluating classifiers, as in supervised learning labels are available and we can compute performance statistics such as accuracy. In clustering we do not have such labels and we can only base our quality estimates on the data and the partition under consideration. Several internal validity measures have been defined that give such quality estimates. A first extensive experimental comparison was performed by Milligan and Cooper [8]. More recently, Arbelaitz *et al*. [2] and Vendramin *et al*. [13] performed similar experiments with an improved methodology [6] and an updated set of validity measures. In these studies the main goal was to evaluate the performance of the validity measures.

In this paper we experiment with four internal validity measures and six clustering algorithms. We want to investigate whether these measures allow for a comparison between algorithms, and whether the discussed strategy of generating several solutions and selecting the best one according to these measures is useful. We do this by comparing validity scores obtained for clusterings produced by the different clustering algorithms on 27 UCI data sets.

2. Validity measures

Until now, we only mentioned *internal* validity measures, as these are the ones that can be directly used in algorithm and parameter selection. Such measures quantify the quality of a clustering relying only on properties intrinsic to the data. Examples include the silhouette, Davies-Bouldin and Caliński-Harabasz measures. A second category consists of the *external* measures, which compare a clustering to a given partition. Examples include the Rand and Jaccard measures.

2.1. External measures

It is important to note that in a typical clustering setting, we cannot rely on external measures to guide us in choosing an appropriate algorithm or good parameter settings, as we do not have a partition to compare to. In contrast, we are trying to find a good partition. However, external measures are often used to evaluate both clustering algorithms and internal validity measures.

A common strategy to evaluate clustering algorithms is to use them to cluster data sets for which a "ground-truth" labelling is known, which are mostly classification data sets. The produced partition is then compared to the known partition using an external index. A high value of the index indicates a good partition, meaning that the clustering algorithm

has succesfully identified the already known structure. As discussed by Färber *et al*. [5], this is often a flawed strategy. One of the reasons is that the class labels generally only indicate one possible grouping of the data set, while for many data sets several clusterings may be useful. This means that a good partition that identifies valid structure different from the given labelling is very likely to score bad on external measures.

A commonly used external measure is the Rand index [11]. It measures the similarity between two partitions of a data set. In the context of external cluster validation one of these two partitions is the reference partition, i.e. the partition denoting the "true" cluster structure. Given two partitions of the data set, P and Q, the Rand Index is defined as follows:

$$RI = \frac{a+d}{a+b+c+d} \tag{1}$$

with

- a = # pairs of elements in the same cluster in P and Q
- b = # pairs of elements in the same cluster in P, but in different clusters in Q
- c = # pairs of elements in a different cluster in P, but in the same cluster in Q
- d = # pairs of elements in a different cluster in both P and Q

$a + d$ can be seen as the number of agreements between the two partitions, whereas $b + c$ can be seen as the number of disagreements. The Rand index is in $[0, 1]$, 1 indicating a perfect match. A problem with this measure is that the expected value for two random partitions is not a constant value. This is one of the reasons why the Adjusted Rand index (ARI) was introduced. It is defined as:

$$ARI = \frac{a - \frac{(a+c)(a+b)}{a+b+c+d}}{\frac{(a+c)+(a+b)}{2} - \frac{(a+c)(a+b)}{a+b+c+d}} \tag{2}$$

The ARI has expected value 0 for random partitions, and still has a maximum value of 1 to indicate perfect agreement.

2.2. Internal measures

Internal validity measures only rely on properties intrinsic to the data set. These measures are mathematical formulations that capture some ideas on what a good clustering should look like. They should allow the comparison of partitions with a different number of clusters. The within-cluster sum of squares, which is minimized by *e.g*. k-means, cannot be used because its value will decrease as the number of clusters increases, and reach the optimal value of zero for a solution in which every point is assigned to each own cluster. In the remainder of this section, we discuss the measures used in the experiments. The first three were selected as common representatives of a wide range of more traditional validity measures (see [2, 13] for an extensive overview). The fourth measure (DBCV) is included as it is quite different from the previous three, as will be clear from the remainder of the paper.

Notation

X: data set to be clustered
x_i: object of X for $i \in \{1, \ldots, N\}$, $N = |X|$
$C = \{c_1, c_2, \ldots, c_K\}$: clustering of the data set into K disjoint sets, s.t. $\cup_i c_i = X$
$\overline{c_k} = \frac{1}{|c_k|} \sum_{x_i \in c_k} x_i$: the centroid of a cluster
$\overline{X} = \frac{1}{N} \sum_{x_i \in X} x_i$: the centroid of the data set

Silhouette measure

The silhouette index (SI) [12] was found to be one of the best performing measures in the extensive comparative experiments by Arbelaitz *et al*. [2] and Vendramin *et al*. [13]. We define $a(x_i, c_j)$ as the average distance of object x_i to all other points in its cluster c_j, $d(x_i, c_j)$ as the average distance of x_i to all points in another cluster c_j, and $b(x_i)$ as the minimum over these distances to all other clusters:

$$a(x_i, c_j) = \frac{1}{|c_j|} \sum\nolimits_{x_k \in c_j} d(x_i, x_k) \text{ with } x_i \in c_j \tag{3}$$

$$d(x_i, c_j) = \frac{1}{|c_j|} \sum\nolimits_{x_k \in c_j} d(x_i, x_k) \text{ with } x_i \notin c_j \tag{4}$$

$$b(x_i) = \min_{c_j \in C \setminus c_k} d(x_i, c_j) \text{ with } x_i \in c_k \tag{5}$$

The silhouette value of a single point is defined as:

$$s(x_i) = \frac{b(x_i) - a(x_i, c_j)}{\max\{b(x_i), a(x_i, c_j)\}} \tag{6}$$

The silhouette value of a partition is the average of these values over all points:

$$SI(C) = \frac{1}{N} \sum_{i=1}^{N} s(x_i) \tag{7}$$

Its complexity is $\mathcal{O}(N^2)$, although a simplified version can be used in which distances between objects and clusters are measured by considering the distance to the cluster centroids, instead of taking the average of the distances to all of its points. The silhouette score of a clustering is in $[-1, 1]$, and should be maximized.

Davies-Bouldin measure

The Davies-Bouldin (DB) measure [4] is defined as follows. With cluster scatter S defined as

$$S(c_k) = \frac{1}{|c_k|} \sum_{x_i \in c_k} d(x_i, \overline{c_k}) \tag{8}$$

the Davies-Bouldin index of a partition is

$$DB(C) = \frac{1}{|C|} \sum_{c_k \in C} \max_{c_l \in C \setminus c_k} \left\{ \frac{S(c_k) + S(c_l)}{d(\overline{c_k}, \overline{c_l})} \right\} \quad (9)$$

Its computational complexity is $\mathcal{O}(N)$ if $K \ll N$, which is usually the case. The Davies-Bouldin score of a clustering is in $[0, +\infty]$ and should be minimized.

Caliński-Harabasz measure

The Caliński-Harabasz (CH) measure [3] is defined as

$$CH(C) = \frac{(N - |C|) * \sum_{c_k \in C} |c_k| \, d(\overline{c_k}, \overline{X})}{(|C| - 1) * \sum_{c_k \in C} \sum_{x_i \in c_k} d(x_i, \overline{c_k})} \quad (10)$$

Its computational complexity is $\mathcal{O}(N)$. The Caliński-Harabasz score of a clustering is in $[0, +\infty]$ and should be maximized.

All three above defined measures are essentially a ratio of cluster compactness (points in the same cluster should be similar) and separation (points in different clusters should be dissimilar). They differ in how these two concepts are defined, and how they are combined. More precisely, the silhouette measure defines cluster compactness based on the pairwise distances between all points in the cluster, and separation based on pairwise distances between all points in the cluster and all points in the closest other cluster. The Davies-Bouldin measure defines compactness based on the distance of points in the cluster to its centroid, and separation based on distances between centroids. The Caliński-Harabasz measure also defines compactness based on the distance of points in a cluster to its centroid, and separation as the distance of the cluster centroid to the data centroid. From their definitions it is clear that these measures have a strong bias towards spherical clusterings. This is illustrated in Figure 1. We generated a set of clusterings with spectral clustering by varying the parameters, and selected the best result from this set according to the different measures. SI, CH and DB fail to select the correct solution. The majority of existing validity measures can be expected to show this behaviour. For example, most measures used by Milligan and Cooper [8], Vendramin *et al.* [13] and Arbelaitz *et al.* [2] (which are the three most extensive comparisons of validity measures available, comparing respectively 30, 30 and 40 measures) share this bias.

Density-Based Cluster Validation

The recently proposed Density-Based Cluster Validation (DBCV) measure [9] is quite different from the three previously discussed ones. DBCV can handle clusters with different densities and shapes, as it is based on the notion of an

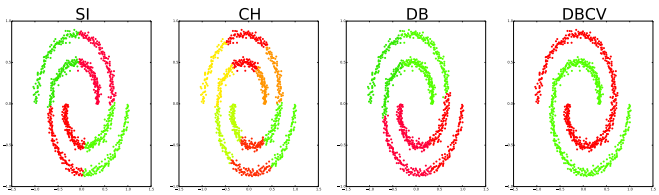

Figure 1: Spectral clustering solutions selected by various measures. Illustration of DBCV being the only internal validity measure that selects the correct solution.

all-points-core-distance ($a_{pts}coredist$) to capture density properties and the use of minimum spanning trees (MSTs) to handle arbitrary shapes. The all-points-core-distance of a point x belonging to cluster c_i is defined as follows

$$a_{pts}coredist(x) = \left(\frac{\sum_{j=2}^{|c_i|} \left(\frac{1}{KNN(x,j)} \right)^d}{|c_i| - 1} \right)^{\frac{-1}{d}} \quad (11)$$

with $x \in c_j$, $KNN(x, j)$ the distance to the j-nearest neighbor of x and d the dimensionality of x. The all-points-core-distance can be seen as the inverse of the density of a point in its cluster. Using the $a_{pts}coredist$, the mutual reachability distance between two objects is defined to incorporate their density properties:

$$d_{mreach}(x_i, x_j) = \max\{a_{pts}coredist(x_i), a_{pts}coredist(x_j), d(x_i, x_j)\} \quad (12)$$

For every cluster a mutual reachability distance graph is constructed, with the cluster points as vertices and de mutual reachability distances as edge weights, and the MST is constructed for each of these graphs. The *density sparseness* of a cluster is then defined as the maximum edge weight of the internal edges of the corresponding MST. The *density separation* of a pair of clusters is defined as the minimum reachability distance between the internal nodes of the MSTs corresponding to each cluster. Based on the density sparseness of a cluster (DSC) and the density separation (DSPC), the validity index of a single cluster is then defined as

$$V_c(c_i) = \frac{\min_{1 \le j \le |C|, j \neq i} \left(DSPC(c_i, c_j) \right) - DSC(c_i)}{\max \left(\min_{1 \le j \le |C|, j \neq i} \left(DSPC(c_i, c_j) \right), DSC(c_i) \right)} \quad (13)$$

And the validity index of a clustering is defined as the weighted average of the validity indices of the clusters

$$DBCV(C) = \sum_{i=1}^{K} \frac{|c_i|}{N} V_c(c_i) \quad (14)$$

Algorithm	Complexity	Parameters
k-means	$\mathcal{O}(NK)$	# K: # clusters
DBSCAN	$\mathcal{O}(Nlog(N))$	ϵ: max. sample dist. to be nbs. $minPts$: # nbs. to be core pt
spectral	$\mathcal{O}(N^3)$	K: # clusters and k: # neighbors to connect or σ: scaling factor of RBF
Ward	$\mathcal{O}(N^2)$	K: # clusters
meanshift	$\mathcal{O}(N^2)$	RBF kernel bandwidth
EM	$\mathcal{O}(NK)$	K: # clusters

Table 1: Algorithms used, their time complexity and parameters

This measure again combines cluster compactness and separation, but these concepts are now defined as properties of graphs built in the transformed space of reachability distances. Its complexity is $\mathcal{O}(N^2)$. The DBCV score of a clustering is in $[-1, 1]$ and should be maximized.

3. Algorithms

Table 1 shows an overview of the algorithms used in the experiments, and the varied parameters. The parameter ranges were chosen to be wide enough to make sure that they contain values leading to a good solution. The algorithms were chosen because they are common representatives of various types of clustering algorithms. They might construct different types of models (*e.g.* k-means produces a set of prototypes, whereas EM produces Gaussian mixture components), but we only consider the hard data set partitions that we can derive from them. We used the scikit-learn [10] implementation for all algorithms.

We perform a grid search over the parameter ranges for each algorithm and data set combination, trying a maximum of 100 parameter combinations. For k-means, Ward and EM all numbers of clusters in the range were tried. For DBSCAN, spectral and meanshift the real-valued parameters were taken to be evenly spaced over the given intervals. Parameter tuning is necessary, as simply using the default parameters usually produces much lower scores than those obtained with the grid search. We want to investigate the ability of these algorithms to produce clusterings that score well on the given validity measures, so we can select the "best" parameter configuration as the one producing the best clustering according to the measure. If we would just be interested in using one clustering algorithm, different strategies to select the algorithm parameters might be more appropriate. For example, Zelnik-Manor and Perona [14] discuss automated ways to set the parameters of spectral clustering, including the number of clusters.

The complexities given in Table 1 apply for one algorithm run. As we need to evaluate the quality of all the clusterings produced in the grid search, the actual complexity of running an algorithm for a particular data set is

$$k(C(A) + C(E)) \tag{15}$$

with k the number of tried parameter combinations, $C(A)$ the complexity of one run of algorithm A and $C(E)$ the complexity of evaluating the quality of one partition using evaluation measure E. For example, if we use k-means in combination with the silhouette index, the resulting time complexity is $\mathcal{O}(k(N + N^2)) = \mathcal{O}(N^2)$. This means that we spend much more time evaluating the clustering than producing it.

4. Results

In this section we investigate the relative abilities of the discussed clustering algorithms to score well on the validity measures. We have performed experiments with 27 UCI data sets (listed in appendix A). An important issue to consider when making such a comparison is the fact that DBSCAN and meanshift are able to identify points as noise, while the other algorithms are not. The points identified as noise can actually be "true" noise, but don't have to be: they are simply the points that agree with the algorithms' definition of noise, under a certain parameter configuration. What should we do with such points in the context of calculating validity measures? Most work on cluster validity does not deal with this issue, as they do not use algorithms that are able to detect noise. As a result there is no agreed upon procedure on how to do this. Moulavi *et al.* [9] however discuss various noise handling strategies in the context of evaluating their DBCV validity measure. Two of the strategies are to (1) assign each noise point to its closest cluster and (2) remove all noise points before calculating the index with a proportional penalty. In the latter strategy, given the set of noise points O, the penalized score S' is determined as $S'(C) = S(C)\frac{N-|O|}{N}$ with C a partition that is not defined on the entire data set, i.e. $(\cup_i c_i) \cup O = X$. Such procedures are needed for SI, CH and DB, but in principle not for DBCV, as this is the only measure that is able to deal with noise without modifications. In the DBCV measure, the validity of a partition is a weighted sum of the validity scores of the individual clusters, with the weights proportional to the cluster sizes. Points identified as noise simply do not contribute to this sum. This reduces the maximal score a clustering can get, and increases the minimal score. For example, if 50% of the points are identified as noise, the DBCV score will be in $[-0.5, 0.5]$ (whereas it is in $[-1, 1]$ if no noise is identified).

In the remainder of this section we discuss some observations that were made during the experiments. We have used the second noise handling strategy (removing noise points and applying a penalty), unless stated otherwise.

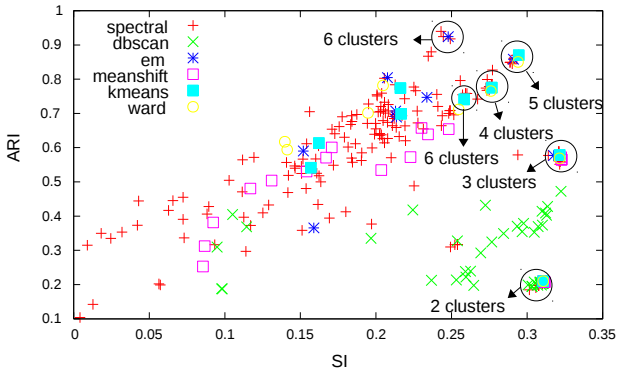

Figure 2: SI vs. ARI for the *dermatology* data set

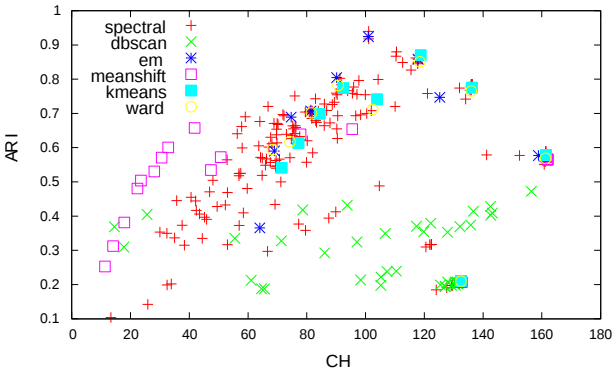

Figure 3: CH vs. ARI for the *dermatology* data set

Assessing algorithms and internal measures using external measures can be misleading. While this is not a new observation [5], comparing with external measures is still a common strategy [2]. One of the reasons why this can be misleading is illustrated in Figures 2 and 3, which show the SI and CH scores versus the ARI for the *dermatology* data set. While most algorithms are able to produce solutions with a relatively high ARI (ca. 0.9), both the SI and CH measures prefer 3-cluster solutions that yield a much lower ARI value over the true 6-cluster partition. In these 3-cluster solutions, some clusters from the 6-cluster solution are merged. The plot indicates that according to the SI and CH measures, 2-, 3-, and 5-cluster solutions can be reasonable partitions of the data set. This suggests that the classes actually form a hierarchy, and that several cuts of the dendogram are sensible to arrive at a good partitional clustering. However, as this is a typical classification data set, we only have one set of class labels to which we can compare. This illustrates just one possible reason why comparisons with external measures in the evaluation of clustering algorithms and internal measures can be very misleading. From a high score on an external measure we can conclude that the algorithm has succesfully recovered the already known structure, but from a low score we cannot conclude that the produced clustering is bad. This is problematic for some common evaluation methodologies, such as examining the correlation between internal and external scores or evaluating the ability of an internal measure to select the solution with the highest ARI.

Highly imbalanced clusterings score well. Figure 4 illustrates that for the *sonar* data set, DBSCAN and meanshift are able to obtain significantly higher SI scores than the other algorithms. Similar behaviour was observed for several other data sets. Closer inspection shows that these high scoring clusterings are all very imbalanced, separating only a few points from all the others. DBSCAN and meanshift are the only algorithms that generated such clusterings.

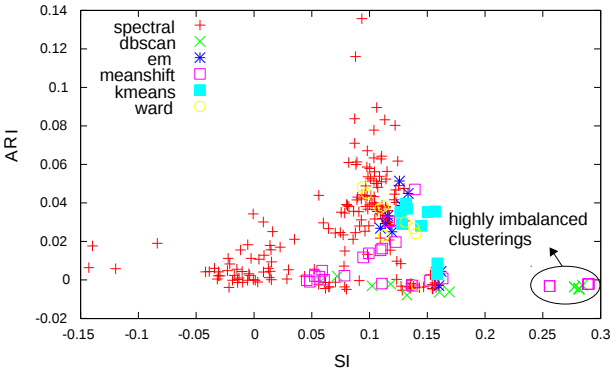

Figure 4: SI vs. ARI for the *sonar* data set

In particular, the strategy of isolating one point and grouping all other points together tends to score well on the silhouette index. This preference for imbalanced clusterings was also observed for other measures, *e.g.* illustrated for DBCV in Figure 6. This seems to be unwanted behaviour for clustering, as we are looking for interesting structure in the data set, and simply separating one or a few points from the others usually does not qualify as such.

All measures are heavily influenced by points identified as noise. Often the increase in the validity score due to identifying some points as noise is larger than the reduction of the score by applying a penalty afterwards. Figure 5 illustrates this for the CH index. The indicated clusterings obtained by DBSCAN are very similar to the 2-cluster solution produced by k-means: disregarding the small set of noise points that is identified by DBSCAN, they identify the same structure. Similar observations were also made for the DBCV index, as illustrated in Figure 6. Identifying a limited set of noise points can greatly increase scores, giving a large advantage to algorithms able to do this (DBSCAN and meanshift in these experiments). This can hide the fact that the identified partitions are often actually very similar. For

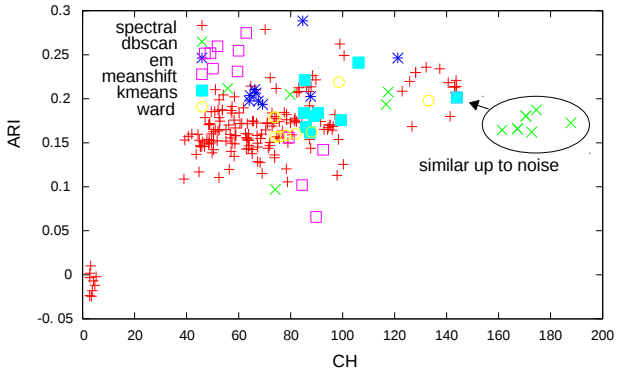

Figure 5: CH vs. ARI for the *glass* data set

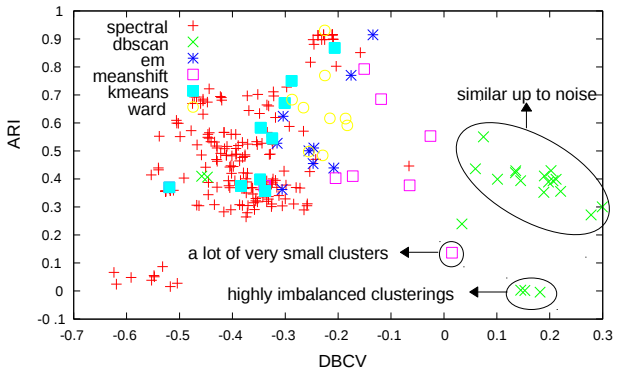

Figure 6: DBDCV vs. ARI for the *wine* data set

the DBCV measure this effect is very severe, as meanshift and DBSCAN outperform all other algorithms on nearly all 27 data sets in its standard setting. While this could be expected because of their similar assumptions about cluster structure, it is actually caused by the above mentioned effect of identifying noise points.

To allow for a more interesting comparison, we also experimented with the strategy of assigning each noise point to its closest cluster before calculating the DBCV score. Figure 7 shows the effect of this strategy for the *wine* data set.

We can simply use k-means to score well on the silhouette and Caliński-Harabasz measures. If we remove highly imbalanced clusterings (defined as the ones with $\frac{|c_{k-1}|}{|c_k|} < 0.1$, with c_k and c_{k-1} the largest and second-to-largest clusters, respectively) and solutions in which more than 50% of the points are identified as noise, most algorithms attain very similar maximal scores for the SI measure. This is illustrated in Table 2, which shows the average relative scores for all algorithms over the 27 data sets. With $v_{\mathrm{a}}^{\mathrm{d}}$ as the best found validity score for data set d by algorithm a, we define the relative score of a particular algorithm for

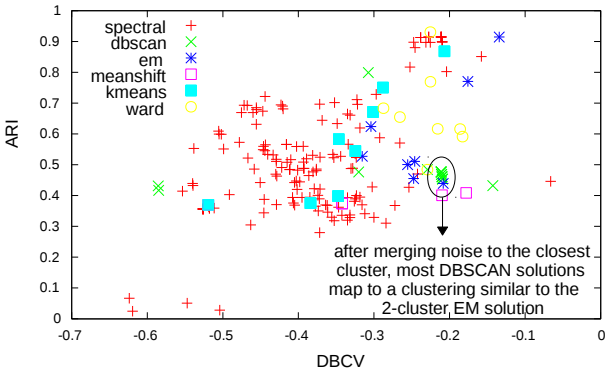

Figure 7: DBCV vs. ARI for the *wine*, after merging noise to the closest cluster and removing imbalanced clusterings

a data set as $\frac{v_a^d}{\max_{x \in \text{algos}} v_x^d}$. All algorithms attain a high average relative score for the SI measure. In particular, spectral and k-means clustering perform well. Similar conclusions hold for the CH score (shown in Table 3). Overall, based on the results on these 27 data sets, it seems reasonable to simply use k-means to produce clusterings with a good SI or CH score. Spectral clustering could also be used as it obtains very similar scores, but at a much higher computational cost.

	SI	
	Mean	Std
spectral	0.97	0.044
k-means	0.97	0.050
Ward	0.92	0.085
meanshift	0.91	0.12
EM	0.88	0.15
DBSCAN	0.83	0.25

Table 2: Average relative SI score over 27 UCI data sets. Noise is penalized, and imbalanced clusterings are filtered.

	CH	
	Mean	Std
k-means	0.96	0.070
spectral	0.95	0.069
Ward	0.87	0.091
EM	0.81	0.20
DBSCAN	0.71	0.32
meanshift	0.63	0.22

Table 3: Average relative CH score over 27 UCI data sets. Noise is penalized, and imbalanced clusterings are filtered.

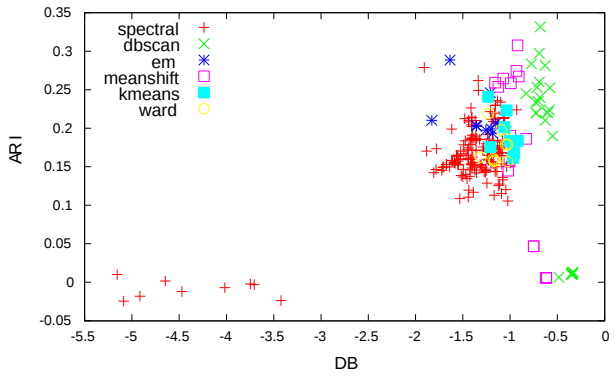

Figure 8: DB vs. ARI for the *glass* data set (DB index multiplied with -1 for easier comparison), very noisy and imbalanced clusterings are filtered.

A similar conclusion does not hold for the DB measure: after removing very noisy and imbalanced clusterings meanshift and DBSCAN are still the best scoring algorithms (shown in Table 4), indicating that the DB measure is more sensitive to points identified as noise. This is illustrated for the *glass* data set in Figure 8. For the DB index lower values are better, so the ranks are now determined as $\frac{\min_{x \in \text{algos}} v_x^d}{v_a^d}$.

	DB	
	Mean	Std
DBSCAN	0.94	0.080
meanshift	0.94	0.22
spectral	0.88	0.12
k-means	0.86	0.13
Ward	0.85	0.14
EM	0.79	0.17

Table 4: Average relative DB score over 27 UCI data sets. Noise is penalized, and imbalanced clusterings are filtered.

For the DBCV measure, the strategy of filtering out imbalanced solutions "manually" with some threshold value does not work well. For the previous measures it were only the few most imbalanced clusterings that scored high, whereas for this measure the amount of imbalance seems much more correlated with the validity score, making the choice of a threshold and the resulting relative scores quite arbitrary. The data sets on which DBCV does yield comparable results seem to be those with clearer structure. This indicates that DBCV can be useful for data sets with well separated structure, but the results become less interesting as data becomes more noisy or transitions between clusters become more blurred.

5. Conclusion

One way to compare clustering algorithms is by using internal validity measures to assess the quality of the clusterings they produce. In this paper we study the behaviour of four such measures on clusterings generated by six very different clustering algorithms. The goal is to provide insights into both the validity measures and the ability of the algorithms to score well on the measures. This could help a user in selecting an appropriate validity measure and clustering algorithm. We conclude that none of the four measures under consideration can be used to make a fair comparison between the six algorithms. All measures exhibit some undesired properties: sensitivity to points identified as noise, a preference for highly imbalanced solutions, or a bias towards spherical clusterings. To produce clusterings that score well on the silhouette and Caliński-Harabasz measures, we can simply use k-means. This does not come as a surprise, as they are based on similar assumptions about cluster structure. To score well on the Davies-Bouldin and DBCV measures, we can use DBSCAN or meanshift, but this is mainly due to the previously mentioned undesired properties.

Acknowledgements

We would like to thank Antoine Adam for many useful suggestions. This work is supported by the Agency for Innovation by Science and Technology in Flanders (IWT).

References

[1] M. Ackerman, S. Ben-David, and D. Loker. Towards property-based classification of clustering paradigms. In J. Lafferty, C. Williams, J. Shawe-taylor, R. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 10–18. 2010.

[2] O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J. M. Pérez, and I. n. Perona. An extensive comparative study of cluster validity indices. *Pattern Recognition*, 46(1):243–256, Jan. 2013.

[3] T. Caliński and J. Harabasz. A dendrite method for cluster analysis. *Communications in Statistics*, 3(1):1–27, 1974.

[4] D. L. Davies and D. W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(2):224–227, 1979.

[5] I. Färber, S. Günnemann, H.-P. Kriegel, P. Kröger, E. Müller, E. Schubert, T. Seidl, and A. Zimek. On Using Class-Labels in Evaluation of Clusterings. In *Proc. MultiClust Workshop in conjunction with 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2010), Washington, DC, USA*, 2010.

[6] I. Gurrutxaga, J. Muguerza, O. Arbelaitz, J. M. Pérez, and J. I. Martín. Towards a standard methodology to evaluate internal cluster validity indices. *Pattern Recognition Letters*, 32(3):505–515, 2011.

[7] A. K. Jain. Data clustering : 50 years beyond K-means. *Pattern Recognition Letters*, 31:651–666, 2010.

[8] G. W. Milligan and M. C. Cooper. An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50(2):159–179, 1985.

[9] D. Moulavi, P. A. Jaskowiak, R. Campello, A. Zimek, and J. Sander. Density-based clustering validation. In *Proceedings of the 14th SIAM International Conference on Data Mining (SDM), Philadelphia, PA*, 2014.

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[11] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.

[12] P. J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

[13] L. Vendramin, R. J. G. B. Campello, E. R. Hruschka, S. Analysis, and D. Mining. Relative clustering validity criteria: A comparative overview. *Statistical Analysis and Data Mining*, 3(4):209–235, 2010.

[14] L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. In *Advances in Neural Information Processing Systems 17*, pages 1601–1608. MIT Press, 2004.

A. UCI data sets

dataset	# samples	# features
bands	277	37
breast cancer Wisconsin	449	9
credit approval	653	15
dermatology	358	34
echocardiogram	87	10
ecoli	336	7
glass	213	9
haberman	289	3
Hayes-Roth	78	4
heart disease (Cleveland)	297	13
heart disease (Switzerland)	105	9
hepatitis	112	18
house votes 84	160	16
Indian liver patient	570	10
ionosphere	350	33
iris	147	4
lenses	24	4
robot failures lp2	46	90
robot failures lp3	46	90
mammographic masses	564	5
movement libras	330	90
post-operative	77	8
sonar	208	59
soybean (small)	47	21
vertebral column 2C	310	6
vertebral column 3C	310	6
wine	178	13

Table 5: Statistics on used UCI datasets.